Test Plan

Introduction

This testing plan outlines the testing approach for the Running Planner application. It describes the scope, test data, approach, entry and exit criteria, coverage and tools needed to ensure quality, functionality and performance.

Scope

In-Scope:

- Account creation and login.
- User authentication and token.
- Training plan creation and scheduling.
- Feedback system.
- Input validation and error handling.
- Data persistence and integrity.

Out-of-Scope:

- UI style, responsiveness and browser compatibility.
- Map and route creation.

Test Data

Test data will include valid and invalid data for all endpoints to test that the endpoints work as intended and can correctly handle errors. Test data will be created for the test environment to ensure that it maintains a consistent state in the test environment.

Testing Approach

The type of testing used in this project will be functional testing to ensure that backend functionality behaves as expected. Testing will be performed using automated and manual testing methods.

Unit testing will focus on individual functions and components.

Integration testing will verify the interaction between the different layers such as the databases, repositories, services and controllers.

System testing will ensure that the backend functions correctly as a whole.

User Acceptance Testing ensures the backend features meet the requirements from the user's point of view.

Entry and Exit Criteria

Entry Criteria:

- Test plan and strategy developed.
- Testable functionality for the backend has been created.

Exit Criteria:

- All test cases pass.
- All bugs found during testing have been fixed.
- Results and metrics have been logged.

Tools

Moq will be used to simulate the backend environment for the tests.

XUnit for writing and running unit, integration and system tests.

Postman for manual testing of API endpoints.

GitHub actions for automated running of unit, integration and system tests.

JMeter for testing the system performance under load.

Coverlet for documenting code coverage.

Test Coverage

There will be at least 70% test coverage of the backend.

Testing to ensure user data is handled correctly and safely will be prioritized.